A Semi-Automatic Approach for Eliciting Cloud Security and Privacy Requirements

Nikolaos Argyropoulos*, Shaun Shei*, Christos Kalloniatis*†, Haralambos Mouratidis*, Aidan Delaney*, Andrew Fish*, Stefanos Gritzalis‡

*School of Computing, Engineering and Mathematics, University of Brighton, Brighton, United Kingdom
{N.Argyropoulos, S.Shei, H.Mouratidis, A.J.Delaney, Andrew.Fish }@brighton.ac.uk
†Department of Cultural Technology and Communication, University of the Aegean, Lesvos, Greece
‡Department of Information and Communication Systems Engineering, University of the Aegean, Samos, Greece
{Chkallon, Sgritz}@aegean.gr

***Abstract*—Cloud computing provides a wide range of services to organisations in a flexible and cost efficient manner. Nevertheless, inherent cloud security issues make organisations hesitant towards the migration of their services to cloud. In parallel, the cloud service-oriented nature requires a specific and more demanding description of the business functional requirements intended for migration. Organisations need to transform their functional requirements based on a specific language, taking into account the respective non-functional requirements of the migrating services. Thus, the need for an approach that will holistically capture organisations' security and privacy requirements and transform them to cloud service requirements is immense. To this end, this paper presents an approach that takes as input abstract security and privacy requirements and produces through a semi-automatic process various alternative implementation options for cloud services. To achieve that a series of model transformations are utilised in order to create a mapping between the organisational and the operational level of the system's analysis.**

***Keywords*—Security, Privacy, Cloud Computing, Semi-automatic Process, Business Process Modelling.**

I. Introduction

Cloud computing is a paradigm where computing resources owned by third party providers are offered as a self-serviceable commodity through the concept of a utility model, accessible through network connections. We derive our definitions of a cloud computing service around the common cloud attributes provided by NIST [12], as well as the ones described in the extended related work. NIST describes the common characteristics for cloud models as; on-demand self-service, ubiquitous network access, rapid elasticity, location independent resource pooling and measured service.

In order to understand and represent the user requirements in terms of enabling organisational strategy to encompass business needs and provide some offering to customers, we need to be able to describe the context of the system. The goals which an organisation aims to achieve by the execution of its business processes can provide highly relevant input during the system's design phase. Goal-oriented requirements engineering (GORE) approaches use goals to capture the rationale behind design-time decisions. Therefore, when paired with business process modelling approaches, they are a useful initial tool for the design of an organisation's processes [6].

In this work we focus on security and privacy oriented aspects of GORE and business process modelling, as our research is primary targeted at organisations seeking to understand the security and privacy impacts of cloud computing on their processes. This typically involves the analysis and transformation of one or more complex systems residing in a business environment to a cloud computing environment. Software systems represent a dynamic environment, where the interaction between different technologies, the exchange of data and the participating actors enable the delivery of business offerings. However cloud computing presents several new security and privacy-related challenges as a result of the various technologies involved, both in terms of traditional established paradigms and the amalgamation of different methods for delivering computing resources [4], [18].

This highlights the multiple layers of abstraction required to understand the complexities in relationships and entities. Thus we need to capture the users security and privacy requirements within the context of their organisational environment. There are many methodologies that can be applied to capture and represent these concepts, but we will focus on adapting a model-driven security approach for eliciting security and privacy requirements and representing, reasoning and addressing the security and privacy issues and impacts on the users software system in a cloud computing environment [21].

Due to the different language used to describe cloud-based software systems at the various levels of abstraction (i.e., organisational, operational, cloud level), a disconnect is created between the high level strategy, the processes and the description of the properties of the services used to implement them. In the context of security and privacy, this disconnect often results in deficient system designs with costly implications for organisations, both financially and in terms of customer trust [15].

Therefore, there is a need for a holistic approach, capable of connecting the different levels of analysis, beginning from highly-abstract organisational goals and leading up to cloud

URI: http://hdl.handle.net/10125/41749
ISBN: 978-0-9981331-0-2

service requirements, creating a mapping between the different levels of abstraction along the way.

To contribute towards that direction, this works aims to answer the following questions:

1) *Which concepts are required to capture cloud service requirements?*
2) *How can we align cloud service level requirements with high-level security and privacy organisational goals?*

To achieve that, first we present a metamodel containing interrelated concepts, capable of describing a system through different levels of abstraction, and next, based on these concepts we present a semi-automatic approach, aiming to bridge the gap between organisational goals and cloud service requirements, focusing on the context of security and privacy. This approach begins from the elicitation of high level security and privacy requirements and implementation mechanisms from organisational goal models. Rules for transforming such models to business processes containing security and privacy constraint and implementing activities are provided next. Finally, using transformation rules for mapping operational level elements to specific cloud service characteristics, cloud service requirements are extracted from the system's process models. Therefore, organisations in need of designing or migrating systems to the cloud can benefit from such an approach, as it can provide guidance throughout the different levels of abstraction in a semi-automated manner while also allows security and privacy related reasoning.

The rest of this work is structured as follows; Section II introduces our metamodel containing the concepts necessary for describing the different levels of system analysis. Section III briefly describes a real life system used for demonstrating the different steps of the proposed approach which are introduced in Section IV. Finally, related work is discussed in Section V and conclusions and future work in Section VI.

II. Concepts

In this section we outline the concepts required for specifying cloud services aligned with the users high level security and privacy requirements. In order to adequately capture the different levels of abstraction required for the systems analysis, a metamodel is introduced in Fig. 1 which combines concepts representing different perspectives. The building blocks of this metamodel are the following:

Goal-oriented requirements engineering concepts: As discussed earlier, GORE is an effective way of capturing high level organisational strategy via the use of actors, goals, resources and dependencies. Since this work focuses on the aspects of security and privacy, we also need to be able to include such concerns at the highest level of analysis. This is why Secure Tropos [14], [13] concepts were selected, as they allow us to perform security and privacy analysis from an organisational perspective.

Business process modelling concepts: Business processes are an effective tool for describing the participants and the flow of activities and information within a system. BPMN 2.0 [16] is the de facto standard business process modelling language, providing a plethora of concepts for capturing the operational perspective of systems. Therefore, it will be used as a part of this approach in order to capture the functional perspective of system analysis.

TABLE I
Comparison of abstraction and granularity for services

Service	**Business Service**	**Cloud Computing Service**
Users	Users	Cloud users
Service provider	Providers	Service providers
Service description	Service description	Service description
-	Functionalities	Capabilities
-	Resources	Resources
-	Dependencies	Dependencies
-	-	Service model
-	-	Deployment model

Cloud service concepts: The description of cloud computing services requires the introduction of a number of concepts able to capture its unique characteristics. It is important that we are able to describe cloud services at an appropriate level of abstraction in order to accurately elicit both their functional and security and privacy related requirements. Later in this section we present a discussion regarding which aspects of cloud services are required for our analysis and we define concepts that allow us to perform it.

There are several factors to be considered for the creation of the metamodel of Fig. 1. The main challenge is to identify which concepts are required in order to capture the necessary information at each level of the analysis. Additionally, another aspect which needs to be considered is how concepts sourcing from different levels of abstraction and granularity can be interrelated in a meaningful way. To overcome such issues, first we identified the relevant aspects of cloud services and created their respective concepts. Next we created transformation rules that link these cloud service related concepts to the business process level. Finally an additional set of rules was introduced for linking the operational level process concepts to the cloud computing level. Therefore, the transformation rules dictated the concepts which needed to be included from each abstraction level, as well as the relationships between them.

A. Cloud Service concepts

We begin by defining services based on different levels of abstraction and granularity as shown in Tab. I, which we argue is required for capturing and reasoning about security and privacy aspects of cloud computing.

Service: The general perception of a service is something of value offered by someone to someone else. For instance, a company offers online shopping as a service to end users. At the abstract level, a service includes at least one category of *users*, a *service provider* and a *service description*. However, we need to provide a more precise definition of what a service is in the context of cloud computing, in order to reason about its security and privacy aspects.

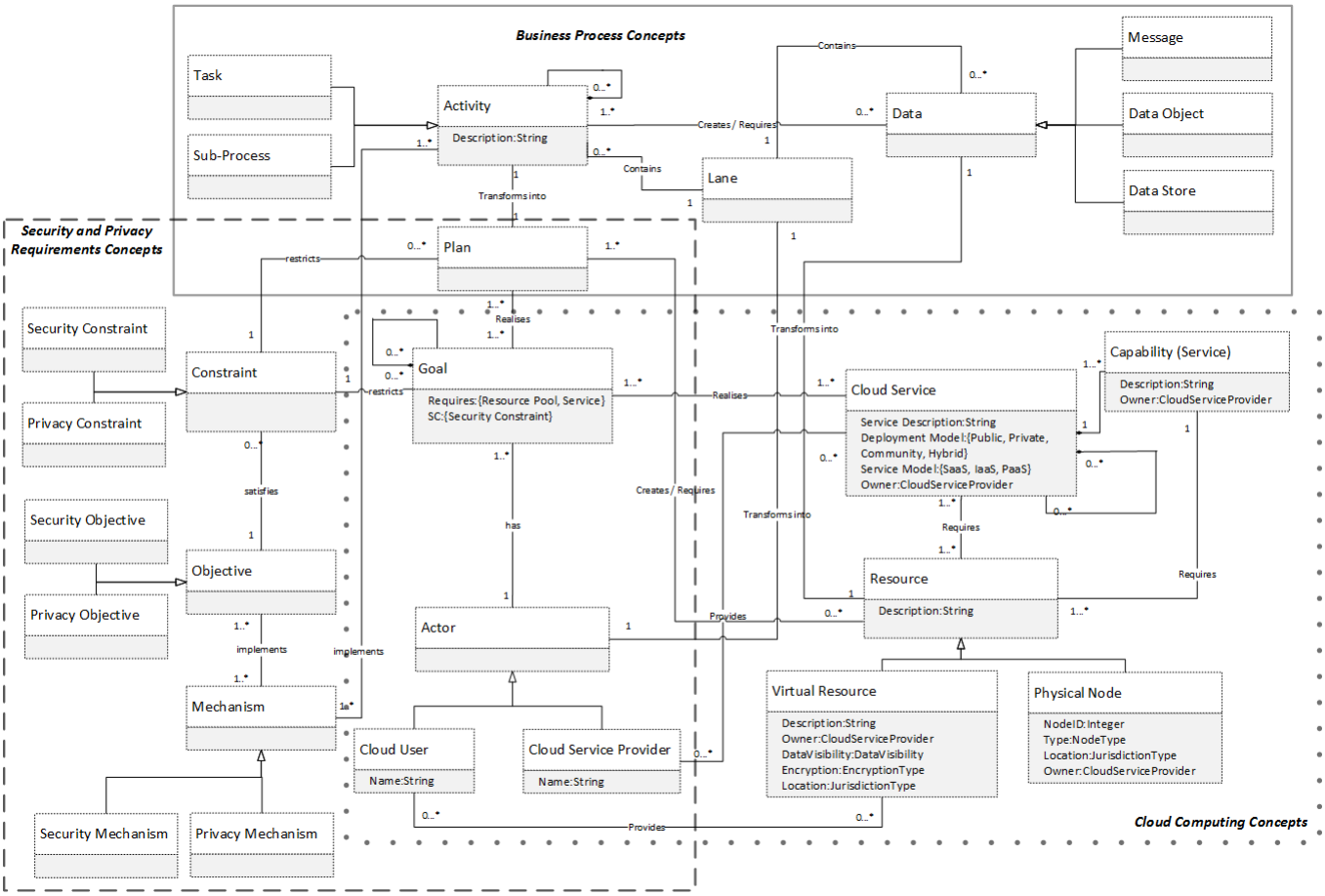

Fig. 1. Metamodel of the proposed approach

Business Service: At a highly abstract business level, services are something of value that is offered to customers. To provide a fine-grained definition of a business service, we provide the following attributes: A *description* of what the service offers, one or many *functionalities*, one or many *users*, one or many *providers*, one or many *resources* and one or many *dependencies*.

Cloud Computing Service: In order to capture cloud computing services, we use the concepts of a business service and extend them through cloud-specific notions, notably the *service model* and *deployment model* attributes. Thus we define a cloud computing service as follows: a cloud service has a *description* of what it offers, one or many *capabilities*, one or many *cloud users*, one or many *service providers*, one or many *resources*, one or many *dependencies*, a *service model* and a *deployment model*. We specify the service and deployment models in order to account for service relationships on the cloud level, i.e. different components which will be managed and physically stored by different providers and reside in varied geographical premises dependent on specified models. Cloud services are delivered through three service models; Software-as-a-Service(SaaS), Platform-as-a-Service(PaaS) and Infrastructure-as-a-Service(IaaS), which can be conceptualised as layers built on top of each other. The deployment model determines the user group, the level of access and the exposure of the cloud service. It also indicates the physical location, ownership and management responsibilities of the computing resources. We capture four deployment levels based on the NIST cloud computing definition; Public, Private, Community and Hybrid.

III. Case Study

This section presents the University of the Aegean Career Office, an existing system, part of which will be used as a case study for the application of our method. The main objective of the University of the Aegean Career Office system is boundary management, i.e. helping students to manage the choices and transitions they need to make upon completing their studies, in order to proceed effectively to the next step of their careers [9]. For the purposes of our example, and due to space limitations of the current work, we will model only a partial view of the system's organisational goals, the conduction of a survey of the university's graduates, performed in order for the university to maintain communication with its graduates. The Career Office creates the survey and outsources its hosting and the gathering

of responses to a cloud service provider. Once the results are collected, they are send back to the Career Office in order to be analysed and be made available to the graduates for discussion. A number of security and privacy issues can be identified when designing such a system, however in order to illustrate a relatively simple example of the application of our approach we will limit our analysis to one security and one privacy constraint at the cloud service provider level.

The context of the Career Office system is fitting with the security and privacy oriented nature of the proposed approach. Nevertheless, since our approach uses elements from two modelling languages (i.e., Secure Tropos and BPMN 2.0) which are flexible and expressive, it is generic enough to be able to support the security and privacy analysis of any system operating in a cloud environment, regardless of the specifics of its context.

IV. Approach

In this section the steps of the proposed approach for eliciting cloud service requirements will be introduced. An overview of the steps of the proposed approach, along with their main inputs and outputs is presented in Fig. 2. Next, each step is individually presented and applied to the Career Office system.

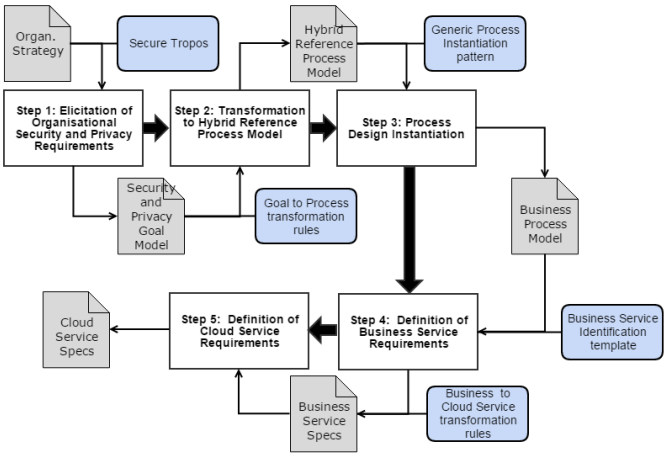

Fig. 2. Overview of the proposed approach

Step 1. Elicitation of Organisational Security and Privacy Requirements. The first step aims to create a goal model of the system to-be by taking into account the overall organisational strategy and the input of system stakeholders. This goal model, consisting of actors, goals, resources and dependencies, will be the main artefact for the identification of the system's security and privacy requirements. For the elicitation of such requirements we use the Secure Tropos approach [14], [13]. Through the concepts of constraints, objectives and mechanisms, offered by the Secure Tropos approach, security and privacy requirements and potential implementation techniques can be captured and linked with the various elements of the goal model. The input of organisational stakeholders and security experts is essential for this step.

In the context of the system selected for our case study, the security and privacy oriented goal model is presented at Fig. 3 where three actors are identified (i.e., Careers office system, Cloud service provider, University graduates). The high level goal of each actor (e.g., "Conduct graduate survey" for the Career office system) is further decomposed to sub-goals (e.g., "Create survey", "Analyse responses" etc.), which can also be decomposed to simpler and less abstract plans (e.g., "Identify survey questions"). Certain actors depend on each other for the achievement of some of their goals (e.g., University graduates depend on the Career Office system for their goal "Participate in graduate survey"), which is captured at the goal model using a dependency relationship. A security constraint regarding the access of the survey form only from the university graduates is identified by the system stakeholders. The identified constraint is connected with the authorisation security requirement, which can be implemented using a number of techniques. In this case candidate security implementation techniques have been identified by security experts as either the use of user credentials (i.e., username and password) or by a user whitelist. Similarly, regarding privacy, a constraint identified dictates that the survey responses cannot be matched to a specific user, which relates to the unlinkability privacy requirement and can be implemented by using various communication protocols (i.e., GAP, TOR) for establishing a private channel between the user and the cloud service.

Step 2. Transformation to Hybrid Reference Process Model. The transformation of the derived goal model to a hybrid reference process model aims to map organisational level security and privacy concerns to operational level activities. Additionally, it captures all the different combinations of possible security and privacy configurations introduced at the goal model level, thus allowing a number of different security and privacy mechanisms to be instantiated when the final business process design is created. Another contribution of this goal-to-process mapping is the adaptability of the produced process designs, as changes at the security requirements of an organisation can, via the hybrid reference model, be traced and reflected to specific parts of its processes and vice versa.

TABLE II
Steps for the goal-to-hybrid reference process model transformation

Phase 1	$\forall ac$ *(actor)* of the goal model: $\exists l(ac)$ *(lane)* in the hybrid model.
Phase 2	$\forall (g\|\|p)$ *(leaf-level goal or plan)* of the goal model: $\exists (a(g)\|\|a(p))$ *(activity)* in the hybrid model.
Phase 3	$\forall r$ *(resource)* of the goal model: $\exists (d(r)\|\|ds(r))$ *(data object or data store)* in the hybrid model.
Phase 4	$\forall c$ *(constraint)*, $\forall o$ *(objective)* and $\forall m$ *(mechanism)* connected to a goal (g), plan (p) or resource (r) of the goal model: *Transfer* it to the hybrid model. *Connect* it to the corresponding activities $(a(g)\|\|a(p))$ or data objects $(d(r)\|\|ds(r))$.

The produced hybrid model includes BPMN 2.0 [16] process elements transformed from Secure Tropos [14] concepts,

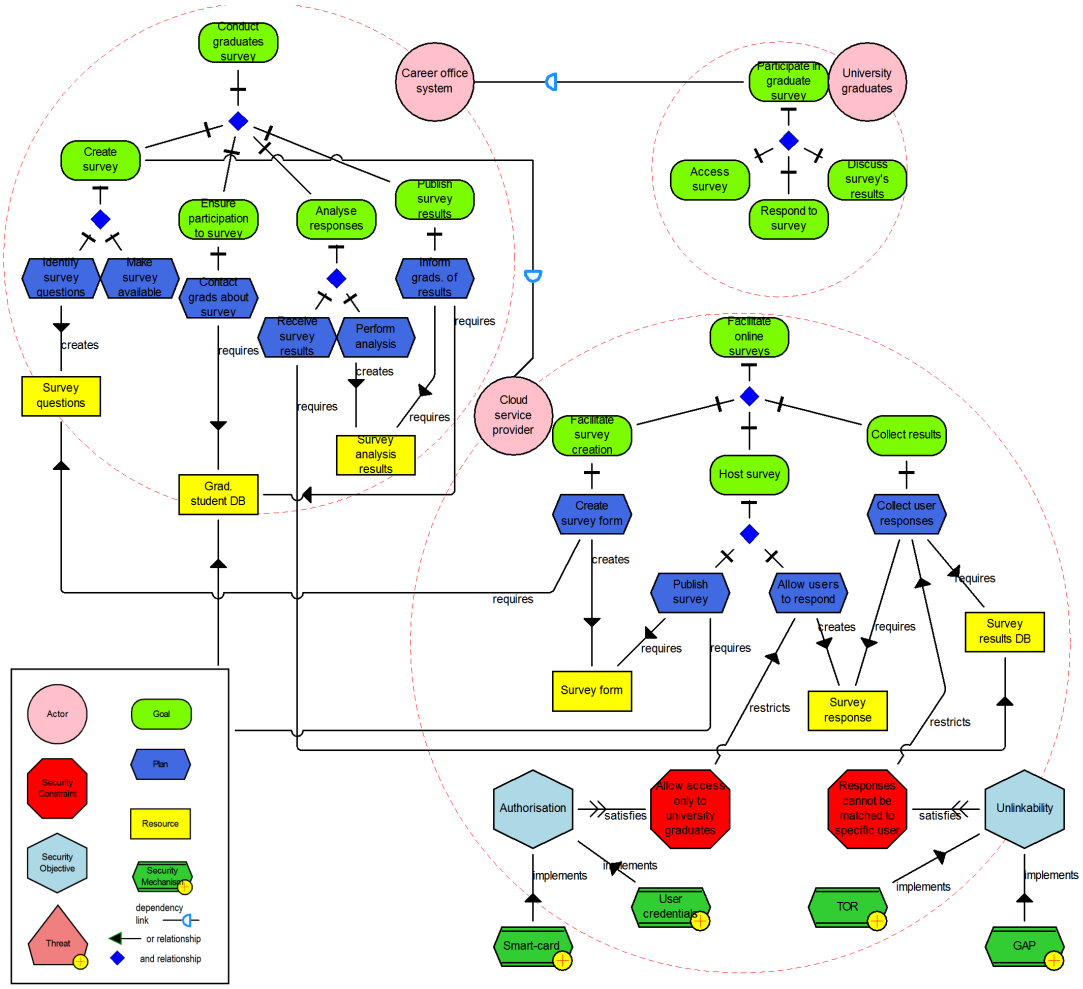

Fig. 3. Partial view of the security and privacy oriented goal model of the Career Office System

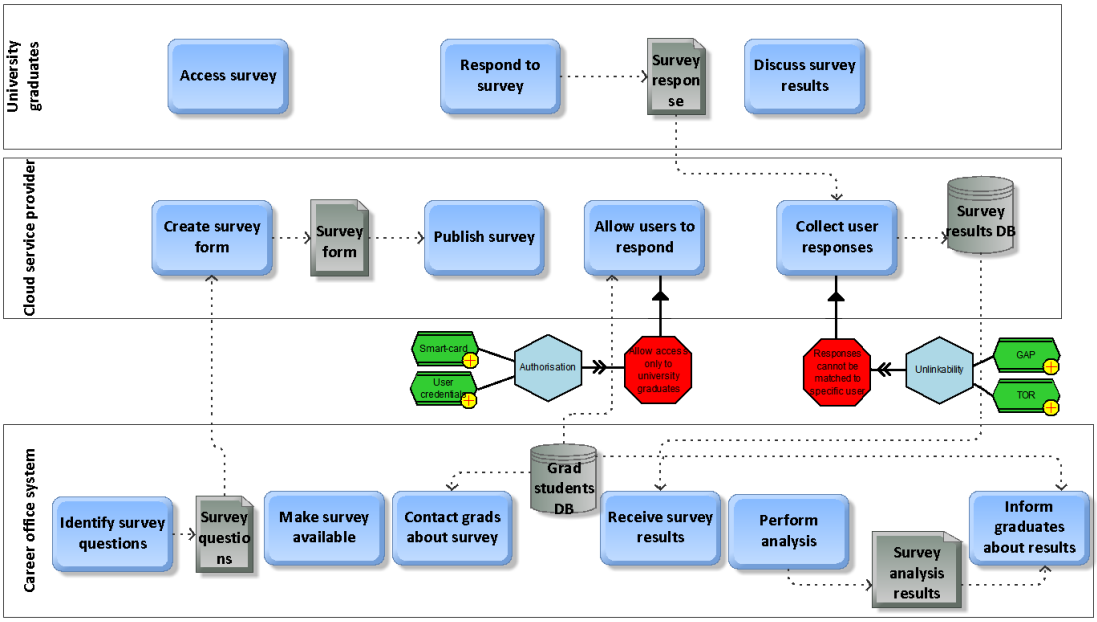

Fig. 4. Hybrid reference process model of the Career Office System

following the transformation rules of Table II, originally introduced in [1]. The security-oriented concepts (constraints, objectives, mechanisms) of the goal model sourcing from Secure Tropos, are directly transferred at the hybrid model, maintaining their connections with the corresponding activities and data objects. The hybrid model of the Career office system, derived by transforming the goal model of Fig. 3 following the above rules, is illustrated in Fig. 4.

Step 3. Process Design Instantiation. In this step, the hybrid model is used to instantiate different secure business process models. The alternatives, in terms of potential security and privacy configurations introduced at the goal model level, create variation points in the hybrid reference model. For each security or privacy constraint activity or resource, different combinations of implementation mechanisms can be selected, according to the specific needs of each process instance. For the operationalisation of such mechanisms we follow a general pattern, introduced in Fig. 5. Using such a pattern, we can automatically interject the operationalisation of the selected security and privacy mechanisms before the constraint activities are performed. When necessary, activities are added at the user and system lane, representing their interaction with the security or privacy implementing activity via the exchange of messages. In order to differentiate the security and privacy related activities at the process level a set of marked padlock symbols are introduced where activities implementing or interacting with mechanisms are marked with an "S" padlock for security and "P" padlock for privacy. Similarly, security constraint activities are denoted with a solid black border while privacy constraint ones are marked with a dashed black border.

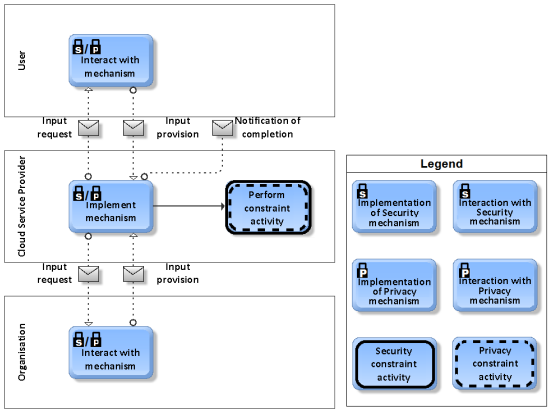

Fig. 5. Process pattern for the operationalisation of security and privacy

In the context of our case study, the produced process model is illustrated in Fig. 6. For its instantiation the security mechanism of "User credentials" and the privacy mechanism of "TOR" have been selected and the appropriate security and privacy implementing activities along with the necessary message exchanges have been introduced at the process model. The creation of control flows has also been manually performed by ordering and connecting the activities and introducing start and end events. It is worth noting that a number of similar but slightly different process models designs could be produced from the hybrid model of Fig. 4, sourcing from a selection of a different set of implementation mechanisms (e.g., "User whitelist" and "GAP") but due to space limitations Fig. 6 presents only one of such possible outcomes.

Step 4. Definition of Business Service Requirements. In order to take advantage of cloud computing offerings, we require well-defined descriptions of the business processes, the user's security and privacy requirements and a strategy for bridging the gap between abstract business goals and cloud service enactments in a cloud environment. In this step we apply the business service template on the process model, in order to generate a list of candidate business services. The template describes the essential characteristics required to capture business activities from a fine-grained perspective, which produces a mapping from the business context to cloud realisation through well-defined service requirements.
We describe each attribute of the business service template and its transformation from the process model in Table III, using an instance generated from the Career Office system as an example. Due to limited space we only show one example of an instantiation, illustrated in Fig. 7, but it is assumed that a complete list of instances has been generated by following the same process.

We define a set of semi-formal rules shown in Table IV, facilitating the transformation of the process model produced in *Step 3* to business service instances. Some attributes offer a range of options, user input is therefore required to generate scenarios according to specific needs.

Fig. 7 shows the instantiation of the "Check Validity of User Credentials" activity, where the user selects the IaaS Service Model and Public as the Deployment Model. This implies at an abstract level that the cloud user will be responsible for managing the activities in all three conceptual layers, where the cloud service provider is only responsible for providing the physical infrastructure. The public model indicates multi-tenancy, meaning that the infrastructure provisioned by the cloud user may also be shared with other tenants.

Step 5. Definition of Cloud Service Requirements. In this step we reason about the need to holistically capture cloud environments, introducing the concepts necessary for the analysis of requirements of cloud systems. Next, using instances of the business service template, generated from the Career Office system, we apply the cloud service requirements model to produce cloud service requirements.

Cloud computing services are abstract notions which require actors, software applications and hardware infrastructure to fully realise. Thus we present the concept of a three layer model to capture the social, application and infrastructure components of a cloud computing environment. This allows us to reason about the interactions between the service and the underlying infrastructure, which is essential for representing and addressing cloud security and privacy requirements. Here we define the concepts required to create a three layer model of a cloud computing environment.

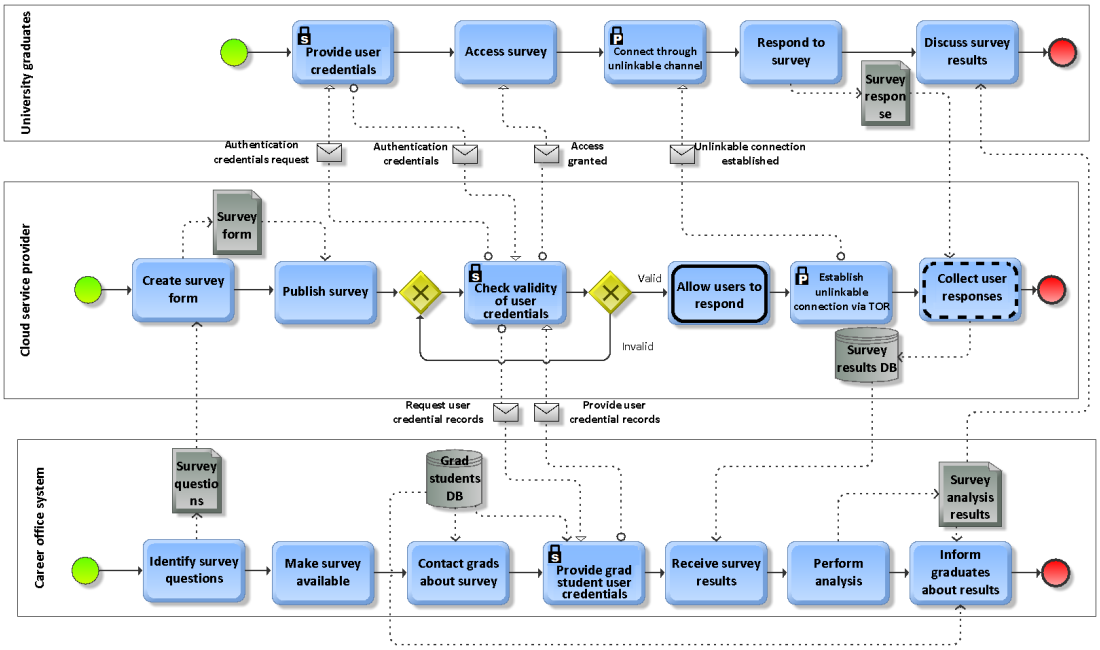

Fig. 6. Business process model of the conduction of an online survey

Activity Description	Check Validity of User Credentials
Functionality	F:(Check Validity of User Credentials, SI, -)
Actors	
	A1: University Graduates A2: Career Office System A3: CSP
Dependencies	
	D1: (Output, A1, R1) D2: (Input, A1, R2) D3: (Output, A1, R3) D4: (Output, A2, R4) D5: (Input, A2, R5)
Security and Privacy Properties	
Security	SI1:(Check Validity of User Credentials, F)
Privacy	-
Resources	
	R1: (VR, Authentication Credentials Request, D1) R2: (VR, Authentication Credentials, D2) R3: (VR, Access Granted, D3) R4: (VR, Request User Credential Records, D4) R5: (VR, Provide User Credential Records, D5)
Service Model	(IaaS)
Deployment Model	(Public)

Fig. 7. Business Service Specification for "Check Validity of User Credentials"

Social Concepts: Based on our definition of a service, a cloud service is provided by a person or organisation. The user of a cloud service is an *actor* representing a person or organisation, who requires computing resources realised through abstraction of virtual and physical resources i.e. *Virtual Resources* or *Physical Nodes* provided by *actors* known as cloud service providers. Therefore we define a conceptual layer to capture all interactions between the users and providers of cloud services, using the concept of *dependencies* where an *actor* depends on another *actor* for the delivery of a *cloud service*. By building a complete view of inter-relationships between users, cloud services and cloud service providers, we are able to identify the accessibility and dependencies of different cloud services between user groups. For example we may generate a scenario where a user group without sufficient access privileges has access to an unauthorised cloud service which modifies confidential company data, in which case the developer is able to amend access controls for the privacy-constrained cloud service. *Application Concepts:* This view abstractly represents the operational view of business processes from the cloud computing perspective, where the context and provision of cloud services are supported through the upper and lower layers respectively. Our definition of a cloud service includes properties required to describe some form of offering which is enabled through computing resources. Thus, each cloud service provides a *description* of its offerings, one or many *capabilities*, *cloud users*, *service providers*, *resources*, a *service model* and a *development model*. A capability is a technical implementation of the ability to perform an action, which from an abstract point of view describes the main goal achieved by the action (e.g., "Implement access control mechanism").

Infrastructure Concepts: The cloud infrastructure forms the foundation, enabling the delivery of cloud computing resources to customers. The physical resources enabling cloud computing services are abstractly decomposed as *processing*, *networking* and *storing* resources on the infrastructure layer, where we model from an abstract perspective, the residency of virtual resources on physical storage. Notions for crucial security and privacy properties such as confidentiality and integrity of assets are captured through these attributes, for example by associating ownership to specific groups of actors or specifying the actors responsible for asset residency. By

TABLE III
ATTRIBUTES OF THE BUSINESS SERVICE TEMPLATE.

Activity Description	This attribute summarises the purpose of the service, which is pulled from the activity name during the transformation (e.g., "Authentication service" or "Check Validity of User Credentials").
Functionality	Each activity is mapped to one function, which indicates the performance of a specific action in order to realise the service (e.g., "Check user request" or "Establish unlinkable connection"). The function also indicates the security and privacy properties associated with the service, in the form of constraints placed on the service or implementation of mechanisms by the service.
Actors	All actors involved in the operation of the service are captured here, which typically involves the end-users, service users and service providers (e.g., "University graduates", "Career Office system", "CSP"). We indicate CSP as a generic cloud service provider since knowledge of specific providers is not required at the requirements level.
Dependencies	We capture the concept of a dependency relationship between actors by providing a description of the dependency, the type of dependency (i.e., input where the depender is implicitly the service provider or output where the dependee is implicitly the service provider), the actors involved in the relationship, and the resource required. For example the dependency of the virtual resource "Authentication credential request" from the depender actor "CSP", which is implicit due to the output dependency type, to the dependee actor "University graduates". Another example for the input dependency type can be seen from the dependence of the virtual resource "Authentication credentials" from the implicit depender actor "CSP" to the dependee actor "University graduates".
Security and Privacy Properties	The security and privacy needs of business services are defined through constraint and security or privacy implementing functionalities: *Security:* The security needs can either be a security constraint on a function or the implementation of a security mechanism by a function e.g. *SI1* indicates that the service "Check validity of user credentials" implements a security mechanism of the same name as the primary functionality. *Privacy:* Similarly for privacy needs, this can be in the form of a privacy constraint on a function or the implementation of a privacy mechanism e.g. *PC1* indicates that the privacy constraint "Collect user responses" is placed on the service of the same name, indicating the constraint on its functionality.
Resources	The type of assets required for delivering the service is specified in order to assign responsibility, map relationships and determine components involved when realising services. For example a required resource can be of type message or document which would be transformed to a virtual resource, or database which corresponds to a physical node e.g. the resource *R4* of type virtual resource "Request user credential records" is required in the dependency *D4*.
Service Model	This attribute represents the level of cloud service the business service is deployed on i.e. SaaS, PaaS or IaaS. User input is required in able to specify the service model they wish to deploy, therefore creating different service model scenarios.
Deployment Model	Similarly the deployment model also presents a range of options to the user, in this case a choice of public, private or hybrid. This also generates different scenarios depending on the model chosen, in addition to the chosen service model i.e. a Public IaaS differs from a Private IaaS as ownership and physical location of resources may be specified differently.

defining these concepts, we are able to model the propagation of service interactions from the abstract social layer down to a more fine-grained enactment of cloud services. Finally, we can capture the consequences of service enactment in the cloud environment.

TABLE IV
EXTRACTION OF BUSINESS SERVICES FROM PROCESS MODEL

Phase 1	**Activity Description:** $\forall an$ *(activity name)* of the process model: $\exists ad(an)$ *(activity description)* in the service template
Phase 2	**Functionality:** $\forall ad$ *(activity description)* of the service template: $\exists f(ad, sc\|\|si\|\|-, pc\|\|pi\|\|-)$ *(functionality, security constraint or security implementation or null, privacy constraint or privacy implementation or null)* in the service template
Phase 3	**Actors:** $\forall ac$ *(actor)* of the process model: $\exists ac(ac)$ *(actor)* in the service template
Phase 4	**Dependencies:** $\forall d$ *(dependency link)* of the process model: $\exists d(i\|\|o, ac(ac), r(r))$ *(input or output and resource)* in the service template
Phase 5	**Security and Privacy Properties:** $\forall f(ad, sc, -), f(ad, si, -), f(ad, -, pc), f(ad, -, pi)$ of the process model: IF $f(ad, sc, -)$ THEN $\exists sc(sc, f)$ *(security constraint)* in the service template IF $f(ad, si, -)$ THEN $\exists si(si, f)$ *(security implementation)* in the service template IF $f(ad, -, pc)$ THEN $\exists pc(pc, f)$ *(privacy constraint)* in the service template IF $f(ad, -, pi)$ THEN $\exists pi(pi, f)$ *(privacy implementation)* in the service template
Phase 6	**Resources:** $\forall do$, $\forall ds$, $\forall m$ *(data object, data store, message)* of the process model: IF$(do\|\|m)$ THEN $\exists r(vr, rd, d(i\|\|o, ac(ac), r(rd))$ *(virtual resource, resource description)* in the service template IF(ds) THEN $\exists r(pn, rd, d(i\|\|o, ac(ac), r(rd))$ *(physical node, resource description)* in the service template
Phase 7	**Service Model:** $\forall sm$ *(service model)* in the service template: $\exists sm(saas\|\|paas\|\|iaas)$ *(SaaS or PaaS or IaaS)* in the service template
Phase 8	**Development Model:** $\forall dm$ *(deployment model)* in the service template: $\exists dm(pu\|\|pr\|\|co\|\|hy)$ *(public or private or community or hybrid)* in the service template

We define a set of semi-formal transformation rules, presented in Table V, that receive business service template instantiations as input and generate a document describing requirements in a cloud environment. The business services are realised as cloud services in the cloud environment model, where the relationships and dependencies are interconnected throughout the three conceptual layers.

Fig. 8 illustrates an instance of the Cloud Environment Template, generated for the Career Office system based on the aggregation of six business service instantiations and applying the transformation rules defined in Table V. In this scenario we discuss the user options and how it impacts the generated output. The user selects four capabilities to aggregate as two separate cloud services. As the capabilities have different service and deployment levels, the generated cloud services are managed at different levels, depending on their individual capabilities. For instance, C3 is IaaS and public, while and C4 is PaaS and private, where the cloud service itself will be split between IaaS and PaaS with the option for selecting who manages the infrastructure resources for the private deployment.

The application of the proposed approach on part of the Career Office system allowed us to get some insight regarding

TABLE V
TRANSFORMATION RULES FROM BUSINESS SERVICE INSTANCES TO THE CLOUD ENVIRONMENT MODEL

< Pre-Conditions >

<**Service Model**> **:**
IF(*SaaS*) THEN ∃ *Social Layer*
IF(*PaaS*) THEN ∃ *Social Layer, Application Layer*
IF(*IaaS*) THEN ∃ *Social Layer, Application Layer, Infrastructure Layer*

<**Deployment Model**> **:**
IF(*Public*) THEN *UserInput*{*Infrastructure Layer, Physical Node*}
IF(*Private*) THEN *UserInput*{*Infrastructure Layer, Physical Node*}
IF(*Hybrid*) THEN *UserInput*{*Infrastructure Layer, Physical Node*}

1. *[All] Map each Business Service instance to a Capability (C1, ... , Cn(Activity Description))* 2. *[User] Option for selecting one or more Capabilities and aggregating it into a Cloud Service instead* 2.1 *[Default behaviour] Maps each unassigned capability to a new cloud service* 3. *[All] Map each actor to Actor list (A1, ... , An)* 4. *[All] Map each Resource to VR if VR, PN if PN ignoring duplicates (VR1, ... , PN1, ... , VRn, PNn)* 5. *[Case-by-case] Map dependency relationship from A1 of CSn to A2 with Rn, where if Input(A1 = CSP), if Output(A2 = CSP)* 6. *[Case-by-case] Map Security and Privacy Properties* 6.1 *Map security properties(Either constraints on capability or capability implements mechanism)* 6.2 *Map privacy properties(Either constraints on capability or capability implements mechanism)* 7. *[Case-by-case] Determine Service Model, following rules above to indicate that CSP is responsible for the layers listed* 8. *[Case-by-case][User] Option for selecting the ownership of Physical Nodes on the infrastructure layer* 9. *[User] Option to generate visual representation of the cloud computing environment*
Social Layer
Actor: An(Actor)

<**Dependency**> **:** Dependency(*A1,CSn(Cn),A2,Rn*)
IF(*Input*) THEN *A1 = CSP*
IF(*Output*) THEN *A2 = CSP*

<**Security and Privacy Properties**>:
IF (*SC || PC*) THEN Constraint(*Type(S || P), Description, Capability*)
IF (*SI || PI*) THEN Mechanism(*Type(S || P), Description, Capability*)

Application Layer

Cloud Service:
IF(*Capability = 1*) THEN CSn(*Capability*)
IF(*Capability > 1*) THEN CSn(*Description[User Option]*)

Capability: IF(*CSn(Cn(Capability), ... , Cn+1(Capability) > 1)*) THEN *CSn(Cn(Capability), ... , Cn+1(Capability))*
Virtual Resource: IF(*Rn = VR*) THEN VRn(*Description, Owner: Actor[User Option]*)
Infrastructure Layer
Physical Node: IF (*Rn =PN*) THEN PNn(*Description, Managed by: Actor || Actor[User Option], Resides on: Actor || Actor[User Option]*)

its added value in real life scenarios. By following the defined steps we were able to produce a set of requirements for a cloud service which can provide to the organisation the required functionality (i.e. conducting an online survey). The level of granularity of the produced cloud service requirements could facilitate the selection of an appropriate cloud service provider by the organisation, as they are able to express their needs in a language closer to the cloud service level. The way such cloud service requirements were produced through our approach, ensures their alignment with the higher levels of abstraction (i.e. organisational and operational). As a result, when a cloud service fulfilling such requirements is selected by the organisation, it can be seamlessly integrated in its system and provide the desired functionality while also complying with its security and privacy needs.

<Pre-Conditions>
2. User selects C1 and C2 to aggregate as one Cloud Service and names it CS1(Survey Service). They also select C4, C5 and names it CS2(User Authentication Service). 2.1 C5 and C6 is automatically converted to a Cloud Service with the same names as the original Capability (CS3, CS4). 3. Duplicate actors are deleted 4. Duplicate resources are deleted 5. Dependency relationships are mapped 6.1 Mechanism(S, C4, C4), Constraint(S, C3, C3) 6.2 Mechanism(P, C5, C5), Constraint(P, C6, C6) 7. ***<Service Model>:*** SaaS(CS1), PaaS(CS2(C4), CS3), IaaS(CS2(C3), CS4) 8. ***<Deployment Model>:*** Public(CS1, CS2(C3), CS3), Private(CS2(C4), CS4), Hybrid(-) 9. User doesn't choose to generate graphical representation
Social Layer
Actor: A1: University Graduates A2: Career Office System A3: CSP
<Dependency>: *D1(A2, CS1(C1), A3, VR1), D2(A3, CS1(C2), A3, VR2), D3(A3, CS2(C1), A1, VR3), D4(A1, CS2(C1), A3, VR4), D5(A3, CS2(C1), A1, VR5), D6(A3, CS2(C1), A2, VR6), D7(A2, CS2(C1), A3, VR7), D8(A3, CS3, A1, VR8), D9(A1, CS4, A3, VR9), D10(A3, CS4, A2, PN1)* ***<Security and Privacy Properties>:*** *SC1(Check validity of user credentials, CS2(C3)), SI1(Allow users to respond, CS2(C4)), PI1(Establish unlinkable connection, CS3), PC1(Collect user responses, CS4)*
Application Layer
Cloud Service: *CS1(Survey Service)* *CS2(User Authentication Service)* *CS3(Establish unlinkable connection)* *CS4(Collect user responses)*
Capability: CS1(C1(Create Survey Form), C2(Publish Survey)), CS2(C3(Check validity of user credentials), C4(Allow users to respond))
Virtual Resource: VR1:(Survey questions, A2), VR2:(Survey form, A3), VR3:(Authentication credentials request, A3), VR4:(Authentication credentials, A1), VR5:(Access granted, A3), VR6:(Request user credential records, A3), VR7:(Provide user credential records, A2), VR8:(Unlinkable connection established, A3), VR9:(Survey responses, A1)
Infrastructure Layer
Physical Node: PN1:(Survey results DB, Managed by: A2, Resides on: A3)

Fig. 8. An instance of a Cloud Environment Model for the Career Office system

V. RELATED WORK

A common approach for providing rationale for design choices at the process level is linking them to organisational strategy via the transformation of goal into process models [19], which is also applicable in the context of security and privacy. For instance, the works in [19], [11], [3] incorporate such an approach in the context of security, while the PriS framework [10] does the same for privacy requirements. In contrast to our approach, the focus of the analysis provided by other works in this area is rather one-dimensional, as they focus on either the social (e.g., interactions between actors and resources) or the technical aspects of security (e.g., selection of services to implement security features). Privacy is also either considered in complete isolation from security, or as just another security requirement. Moreover, their output lacks flexibility as, in case the produced ad-hoc process models need to be slightly modified, the redesign effort needs to begin from scratch, at the goal model level.

Due to the broad range of technologies embodied through the cloud computing paradigm, we survey work from domains such as Service-Orientated Architecture, Web 2.0 and distributed computing to provide common characteristics towards a cloud environment definition [20], [5], [2]. There are numerous efforts in producing a standard cloud computing definition, which is critical in enabling in-depth discussions

and reasoning around cloud characteristics and requirements. Youseff et al. adapts an ontological approach to define cloud computing through different stacks and layers [22]. Qian et al. briefly talks about the categories and standardisation in cloud computing [17]. Jadeja et al. discusses the concepts and challenges in cloud computing [8]. Iankoulova et al. identified nine sub-factors in cloud computing security requirements, where the least researched security areas are *non-repudiation*, *physical protection*, *recovery* and *prosecution* [7]. Zissis et al. systematically capture dynamic characteristics to satisfy security and privacy requirements through trust-based concepts [23]. Schmidt presents a service-oriented approach to facilitate the management of enterprise architectures [20]. Our work addresses the need for a holistic approach to align existing security and privacy properties through a requirements perspective, from a business-oriented view towards concepts on a cloud-specific domain.

VI. Conclusion

This work introduced a semi-automatic approach for the derivation of cloud service requirements originating from high level goals, focusing on security and privacy. A series of model transformations and concept mappings between different modelling languages facilitated the transition between the different levels of abstraction, in order to bridge the gap between highly abstract organisational goals and cloud service requirements. Despite the semi-automated nature of the approach, user input is still required throughout its steps. This is due to the built-in flexibility of certain aspects of the approach, which allows the stakeholders to take informed decisions that will shape the final output according to their specific needs. For instance, at the process instantiation level, a number of business process designs can be extracted from a single hybrid reference model, depending on the selection of security and privacy implementing mechanisms. Similarly, during the business service instantiations stakeholders are able to select the matching of system functionalities to cloud services along with the service and deployment model that best fits their needs. These options are reflected in the generation of requirements for cloud services during the final step of the process. There the stakeholders are presented with a holistic view that provides a more fine-grained representation of how their needs are realised in a cloud computing environment and which is the impact of their choices.

Future work will focus on the creation of better defined process patterns for each type of security and privacy requirement, which can be used to further automate the business process instantiation step. Additionally, concepts from commonly identified domains in cloud computing literature will be introduced, in order to demonstrate and model the impact and propagation of cloud-specific threats and vulnerabilities based on service model and deployment model. Finally, automated support tools will be developed, which by executing the defined transformation rules and receiving user input will be able to implement each step of the proposed approach.

References

[1] N. Argyropoulos, H. Mouratidis and A. Fish, "Towards the Derivation of Secure Business Process Designs", Advances in Conceptual Modelling: ER 2015 Workshops, pp. 248–258, 2015.

[2] A. Erradi, S. Anand and N. Kulkarni, "SOAF: An Architectural Framework for Service Definition and Realization", 2006 IEEE International Conference on Services Computing (SCC'06), 2006.

[3] G. Frankova, M. Séguran, F. Gilcher, S. Trabelsi, J. Dörflinger and M. Aiello, "Deriving business processes with service level agreements from early requirements"', Journal of Systems and Software, vol. 84, no. 8, pp. 1351–1363, 2011.

[4] S. Goyal, "Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review", International Journal of Computer Network and Information Security, vol. 6, no. 3, pp. 20–29, 2014.

[5] C. Höfer and G. Karagiannis, "Cloud computing services: taxonomy and comparison", J Internet Serv Appl, vol. 2, no. 2, pp. 81–94, 2011.

[6] J. Horkoff, T. Li, F. Li, M. Salnitri, E. Cardoso, P. Giorgini, J. Mylopoulos and J. Pimentel, "Taking goal models downstream: A systematic roadmap", 2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS), pp. 1–12, 2014.

[7] I. Iankoulova and M. Daneva, "Cloud computing security requirements: A systematic review", 2012 Sixth International Conference on Research Challenges in Information Science (RCIS), 2012.

[8] Y. Jadeja and K. Modi, "Cloud computing - concepts, architecture and challenges", 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), 2012.

[9] C. Kalloniatis, E. Kavakli and S. Gritzalis, "Dealing with privacy issues during the system design process", Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, pp. 546–551, 2005.

[10] C. Kalloniatis, E. Kavakli and S. Gritzalis, "Addressing privacy requirements in system design: the PriS method", Requirements Engineering, vol. 13, no. 3, pp. 241–255, 2008.

[11] H. A. López, F. Massacci and N. Zannone, "Goal-equivalent secure business process re-engineering", Service-Oriented Computing - ICSOC 2007 Workshops, pp. 212–223, 2009.

[12] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, 2009.

[13] H. Mouratidis, N. Argyropoulos, and S. Shei, "Security Requirements Engineering for Cloud Computing: The Secure Tropos Approach.", Domain-Specific Conceptual Modeling, pp. 357–380, Springer International Publishing, 2016.

[14] H. Mouratidis and P. Giorgini, "Secure Tropos: A security-oriented extension of the Tropos methodology", Int. J. Soft. Eng. Knowl. Eng., vol. 17, no. 02, pp. 285–309, 2007.

[15] T. Neubauer, M. Klemen and S. Biffl, "Secure business process management: a roadmap", First International Conference on Availability, Reliability and Security (ARES'06), 2006.

[16] Object Management Group, "Business Process Model and Notation (BPMN) Version 2.0", 2011.

[17] L. Qian, Z. Luo, Y. Du and L. Guo, "Cloud Computing: An Overview", Lecture Notes in Computer Science, pp. 626–631, 2009.

[18] C. Rong, S. Nguyen and M. Jaatun, "Beyond lightning: A survey on security challenges in cloud computing", Computers & Electrical Engineering, vol. 39, no. 1, pp. 47–54, 2013.

[19] M. Salnitri, E. Paja and P. Giorgini, "From Socio-Technical Requirements to Technical Security Design: an STS-based Framework", DISI-University of Trento, 2015.

[20] R. Schmidt, "A Service-System Based Identification of Meta-services for Service-Oriented Enterprise Architecture", 2011 IEEE 15th International Enterprise Distributed Object Computing Conference Workshops, 2011.

[21] S. Shei, L. Márquez Alcañiz, H. Mouratidis, A. Delaney, D. G. Rosado and E. Fernández-Medina, "Modelling secure cloud systems based on system requirements", Evolving Security and Privacy Requirements Engineering (ESPRE), 2015 IEEE 2nd Workshop on, Ottawa, ON, 2015, pp. 19–24.

[22] L. Youseff, M. Butrico and D. Da Silva, "Toward a Unified Ontology of Cloud Computin", 2008 Grid Computing Environments Workshop, 2008.

[23] D. Zissis and D. Lekkas, "Addressing cloud computing security issues", Future Generation Computer Systems, vol. 28, no. 3, pp. 583–592, 2012.